Unit Testing Plan

for Network Printer System

- **Test Plan**
- **Test Design Specification**
- **Test Cases Specification**

Project Team

T1 Team

Date

2015-11-10

Team Information

하 지 용(201011375)

유 치 영(201211823)

허 인 석(201213220)

라 가 영(201214262)

Table of Contents

1 Introduction

1.1 Objectives

본 문서는 2015년도 2학기 소프트웨어 공학 개론 수업의 T1 Team이 개발한 Network Print System(이하 NPS)을 Unit Testing하기 위한 계획문서이다. T1 Team가 정의한 Unit Testing을 수행하기 위하여 Testing Pass/Fail Criteria를 정의하고 이를 수행하기 위한 Test design & Test cases를 제작한다.

1.2 Background

Network Prineter System(이하 NPS)은 네트워크 환경에서 다수의 유저가 공용 프린터를 이용하여 출력할 수 있는 환경을 소프트웨어적으로 구현한 시스템이다.

Unit test는 시스템을 구성하는 최소 단위 모듈들을 대상으로 하는 test이며, 시스템의 성 능을 좌우하는 요소들이 요구사항을 만족하는지를 확인할 수 있는 기본적인 Test approach이다

1.3 Scope

NPS에 대한 unit test를 수행하기 위한 자원과 절차, test approach와 technique과 필요로 하는 환경 및 도구 등을 정의한다. unit test는 시스템 핵심 기능 관련 프로세스에 중점을 두며, 전달 역할 등 단순 프로세스는 test에서 제외한다.

1.4 Project plan

1.5 Configuration management plan

1.6 References

T1-2015.NPS.SRA-1.3

T1-2015.NPS.SDS-1.1

2 Test items

T1 Team이 SASD기법을 이용하여 개발한 NPS를 Testing한다. SA 에서 최소단위의 각 process 별로 요구사항을 만족하는지, 정상적인 결과가 나오는지, 잘못된 값 입력 시 예외 처리가 동작하는지 Testing을 수행한다. <Figure 1 Overall DFD>은 SA를 이용하여 요구사항을 분석 한 결과를 DFD를 이용하여 나타낸 그림이다. <Figure 2 Structural Chart>은 SD의 Basic Structural Chart를 나타낸 그림이다. 각 그림을 참조하여 Unit을 지정하고, 지정한 Unit을

SRA에 명세 된 내용과 같은 동작을 하는지 확인한다.

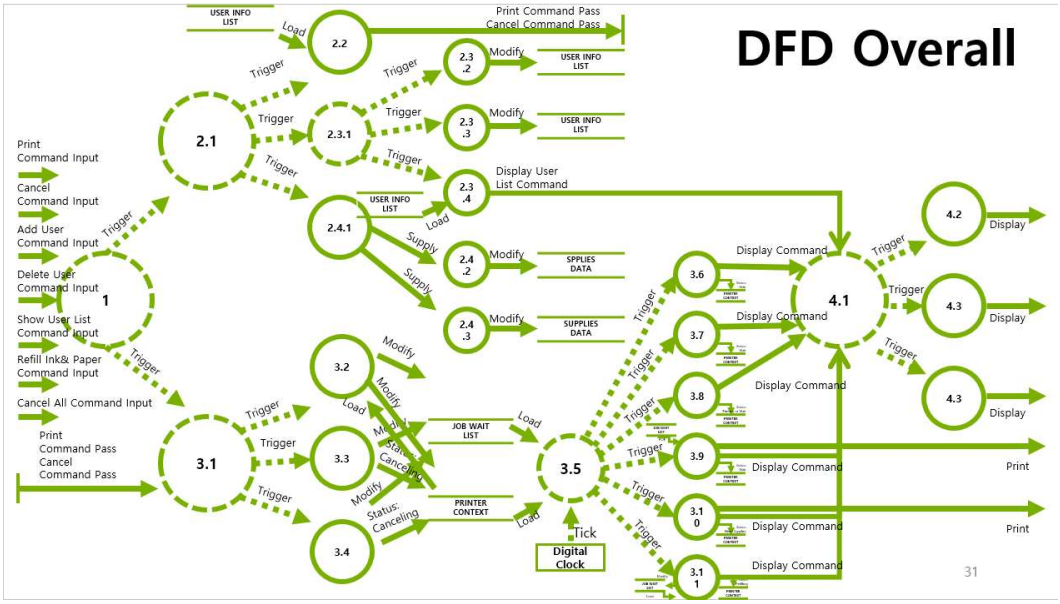

Figure 1 DFD Overall

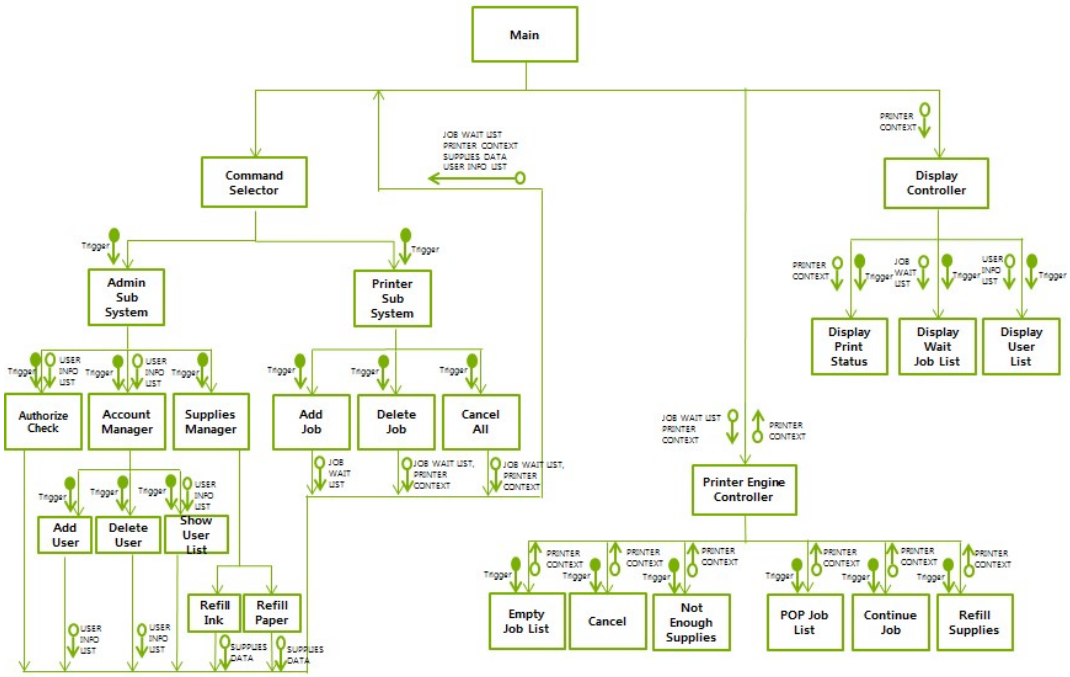

Figure 2 Structural Chart

3 Features to be tested

시스템 핵심 기능 관련 모듈 별로 입력부분과 출력부분을 거쳐 요구사항 만족, 정상적인 동작과 잘못된 값 입력 시 동작에 중점을 두어 테스트한다.

1) Process in SRA : 각 프로세스가 가지고 있는 요구사항을 만족하는지 Test한다.

2) Modules in SDS : 각 모듈이 가지고 있는 데이터 인터페이스를 Test한다.

4 Features not to be tested

전달 역할 등 단순한 프로세스 및 UI구성을 담당하는 프로세스는 테스트에서 제외한다.

5 Approach

NPS의 Program source code 및 Unit Tests는 Visual studio 2010 환경에서 이루어지며, Program code의 변경 및 수정사항은 지속적으로 통합되고 Test된다.

6 Item pass/fail criteria

각 Unit별 Pass / Fail Criteria는 <Table2 Test Case Identification>을 참조한다.

7 Unit test design specification

7.1 Test design specification identifier

7.2 Features to be tested

7.3 Approach refinements

각 Process Specification에 명시된 내용을 기반으로 Test Design 및 Test Cases를 생성해 낸다.

7.4 Test identification

Identifier	Feature
NPS.UTC.1000	입력된 명령을 분류하여 각 명령을 Admin Sub System과 Print Sub System으로 전달한다.
NPS.UTC.2031	관리자가 유저를 추가/삭제 및 유저리스트 확인을 할 것인지를 Add User, Delete User, Show User List로 분류하여 각 프로세스를 실행한다.
NPS.UTC.2032	유저를 추가한다.

NPS.UTC.2033	유저를 삭제한다.
NPS.UTC.2034	유저 리스트를 확인한다.
NPS.UTC.2041	유저가 Ink를 리필 할 것인지, Paper를 리필 할 것인지Supplies Data를 Ink와 Paper로 분류하여 각 프로세스를 실행한다.
NPS.UTC.2042	유저가 Ink를 리필 할 경우, Supplies Data 내 충전할 Ink량을 수정한다.
NPS.UTC.2043	유저가 Paper를 리필 할 경우, Supplies Data 내 충전할 Paper량을 수정한다.
NPS.UTC.3010	명령이 추가 / 삭제, 전체 취소 인지를 확인하며 Add Job, Delete Job, Cancel All로 분류하여 각 프로세스를 실행한다.
NPS.UTC.3020	Admin Sub System에서 authorize check가 된 인쇄 추가 명령을 받아 Job Wait List과 Print Context를 수정한다.
NPS.UTC.3030	Admin Sub System에서 인쇄 삭제 명령을 받아 Job Wait List과 Print Context를 수정한다.
NPS.UTC.3040	Command Parser에서 Cancel All Command를 받아 Job Wait List과 Print Context를 수정한다.
NPS.UTC.3050	JOB WAIT LIST와 PRINTER CONTEXT의 데이터를 받아와 각 프로세스로 전달한다
NPS.UTC.3060	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받아 대기중인 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Wait으로 변경한다.
NPS.UTC.3070	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받아 취소 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Wait로 변경한다.
NPS.UTC.3080	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받아 충전이 요구되는 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Need Supplies로 변경한다.
NPS.UTC.3090	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받고, JOB WAIT LIST에서 최상위 작업을 불러와 출력을 시작한다. 출력 중인 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Printing으로 변경한다
NPS.UTC.3100	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받고, 출력중인 작업을 지속한다.
NPS.UTC.3110	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받고, 잉크와 용지를 SUPPLIES DATA에서 Paper가 0이 아닐 때 10만큼, Ink가 0이 아닐 때 100만큼 감소 후 같은 량을PRINTER CONEXT의 Ink와 Paper에 증가시킨다.

Table 1. Test design Identification

7.5 Feature pass/fail criteria

NPS의 각 모듈(프로세스)은 SRA에 정의되어 있는 요구사항(입력/출력, 동작)을 모두 만족해야 한다. 각 모듈(프로세스)의 입력/출력 및 동작은 SRA의 Process description 항목 및 State Transition Diagram을 참조한다.

8 Unit test case specification

8.1 Test case specification identifier

Identifier	Feature	Pass / Fail Criteria
NPS.UTC.1000	입력된 명령을 분류하여 각 명령을 Admin Sub System과 Print Sub System으로 전달한다.	
NPS.UTC.1000.00	user add user a1	ERROR:not supported command
NPS.UTC.1000.01	user del user a1	ERROR:not supported command
NPS.UTC.1000.02	user show user list	ERROR:not supported command
NPS.UTC.2031	관리자가 유저를 추가/삭제 및 유저리스트 확인을 할 것인지를 Add User, Delete User, Show User List로 분류하여 각 프로세스를 실행한다	
NPS.UTC.2031.00	전달 역할 등 단순한 프로세스 및 UI구성을 담당하는 프로세스는 테스트에서 제외	
NPS.UTC.2032	유저를 추가한다.	
NPS.UTC.2032.00	AddUser("a1")	g_UserInfoList={"a1"}
NPS.UTC.2032.01	g_UserInfoList={"a1"} AddUser("a1")	ERROR:a1 is Already registed.
NPS.UTC.2032.02	AddUser("a11111111111111111111111111111111111111111111")	ERROR:Incorrect UserName
NPS.UTC.2032.03	AddUser(" ")	ERROR:Incorrect UserName
NPS.UTC.2032.04	AddUser("a1 a2")	g_UserInfoList={"a1 a2"}
NPS.UTC.2032.05	AddUser("&a1")	g_UserInfoList={"&a1"}
NPS.UTC.2033	유저를 삭제한다.	
NPS.UTC.2033.00	g_UserInfoList={"a1"} DelUser("a1")	g_UserInfoList==0
NPS.UTC.2033.01	g_UserInfoList={"a1"} DelUser("a2")	ERROR:a2 is not registered
NPS.UTC.2034	유저 리스트를 확인한다.	
NPS.UTC.2034.00	전달 역할 등 단순한 프로세스 및 UI구성을 담당하는 프로세스는 테스트에서 제외	
NPS.UTC.2041	유저가 Ink를 리필 할 것인지, Paper를 리필 할 것인지Supplies Data를 Ink와 Paper로 분류하여 각	

	프로세스를 실행한다.	
NPS.UTC.2041.00	전달 역할 등 단순한 프로세스 및 UI구성을 담당하는 프로세스는 테스트에서 제외	
NPS.UTC.2042	유저가 Ink를 리필 할 경우, Supplies Data 내 충전할 Ink량을 수정한다.	
NPS.UTC.2042.00	KEYWORD_INK == TRUE inkValue =100	g_SuppliesData.Ink={100}
NPS.UTC.2042.01	KEYWORD_INK == TRUE inkValue = -100	ERROR:Invalid Input ink Value
NPS.UTC.2043	유저가 Paper를 리필 할 경우, Supplies Data 내 충전할 Paper량을 수정한다.	
NPS.UTC.2043.00	KEYWORD_PAPER == TRUE paperValue =100	g_SuppliesData.Paper={100}
NPS.UTC.2043.01	KEYWORD_PAPER == TRUE paperValue = -100	ERROR:Invalid Input paper Value
NPS.UTC.2043.02	KEYWORD_PAPER == TRUE paperValue = 100 KEYWORD_INK == TRUE inkValue = 100	g_SuppliesData.Paper={100} g_SuppliesData.Ink={100}
NPS.UTC.2043.03	KEYWORD_PAPER == TRUE paperValue = 100 KEYWORD_INK == TRUE inkValue = -100	g_SuppliesData.Paper={100} ERROR:Invalid Input ink Value
NPS.UTC.2043.04	KEYWORD_PAPER == TRUE paperValue = -100 KEYWORD_INK == TRUE inkValue =100	ERROR:Invalid Input paper Value g_SuppliesData.Ink={100}
NPS.UTC.2043.05	KEYWORD_PAPER == TRUE paperValue = -100 KEYWORD_INK == TRUE inkValue = -100	"ERROR:Invalid Input paper Value ERROR:Invalid Input ink Value
NPS.UTC.2043.06	KEYWORD_INK == TRUE inkValue =100 KEYWORD_PAPER == TRUE paperValue =100	g_SuppliesData.Ink={100} g_SuppliesData.Paper={100}
NPS.UTC.2043.07	KEYWORD_INK == TRUE inkValue =100 KEYWORD_PAPER == TRUE	g_SuppliesData.Ink={100} ERROR:Invalid Input

	paperValue = -100	paper Value
NPC.UTS.2043.08	KEYWORD_INK == TRUE inkValue = -100 KEYWORD_PAPER == TRUE paperValue =100	ERROR:Invalid Input ink Value g_SuppliesData.Paper={100}
NPC.UTS.2043.09	KEYWORD_INK == TRUE inkValue = -100 KEYWORD_PAPER == TRUE paperValue = -100	ERROR:Invalid Input ink Value ERROR:Invalid Input paper Value
NPC.UTS.3010	명령이 추가 / 삭제, 전체 취소 인지를 확인하며 Add Job, Delete Job, Cancel All로 분류하여 각 프로세스를 실행한다.	
NPC.UTS.3010.00	전달 역할 등 단순한 프로세스 및 UI구성을 담당하는 프로세스는 테스트에서 제외	
NPC.UTS.3020	Admin Sub System에서 authorize check가 된 인쇄 추가 명령을 받아 Job Wait List과 Print Context를 수정한다.	
NPC.UTS.3020.00	AddJob(userIdx, "test.txt");	g_JobWaitList={test.txt}
NPC.UTS.3020.01	AddJob(userIdx1, "test1.txt"); AddJob(userIdx2, "test2.txt");	g_JobWaitList={test.txt1 ,test.txt2}
NPC.UTS.3020.02	GetNodeCount(g_JobWaitList)==5 AddJob(userIdx, "test.txt");	ERROR:JOB WAIT LIST FULL
NPC.UTS.3020.03	AddJob(userIdx, "test.txt"); test.txt != EXIST	ERROR:request file open failed
NPC.UTS.3030	Admin Sub System에서 인쇄 삭제 명령을 받아 Job Wait List과 Print Context를 수정한다	
NPC.UTS.3030.00	AddJob(userIdx1, "test1.txt"); AddJob(userIdx2, "test2.txt"); DeleteJob(userIdx1);	g_JobWaitList={test.txt2}
NPC.UTS.3030.01	GetNodeCount(g_JobWaitList)==0 DeleteJob(userIdx);	ERROR: UID : 0 Job not found
NPC.UTS.3030.02	AddJob(userIdx1, "test1.txt"); DeleIteJob(userIdx1);	g_PrinterContext.pCurrentJob=Printing g_PrinterContext.pCurrentJob=NULL
NPC.UTS.3030.03	AddJob(userIdx1, "test1.txt"); AddJob(userIdx2, "test2.txt"); AddJob(userIdx1, "test2.txt");	g_JobWaitList={test.txt2}

	DeleIteJob(userIdx1);	
NPC.UTS.3030.04	g_PrinterContext.status == Printing DeleteJob(currentUserIdx);	g_PrinterContext.status =print g_PrinterContext.status =wait
NPS.UTS.3040	Command Parser에서 Cancel All Command를 받아 Job Wait List과 Print Context를 수정한다.	
NPS.UTS.3040 00	AddJob(userIdx1, "test1.txt"); AddJob(userIdx2, "test2.txt"); AddJob(userIdx3, "test3.txt"); CancelAll();	g_PrinterContext.status = CancelSignal DeleteNode(&g_JobWaitList, pJobNode)
NPC.UTS.3050	JOB WAIT LIST와 PRINTER CONTEXT의 데이터를 받아와 각 프로세스로 전달한다	
NPC.UTS.3050.00	전달 역할 등 단순한 프로세스 및 UI구성을 담당하는 프로세스는 테스트에서 제외	
NPC.UTS.3060	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받아 대기중인 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Wait으로 변경한다.	
NPC.UTS.3060.00	Interfaces는 testing 제외	
NPC.UTS.3070	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받아 취소 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Wait로 변경한다.	
NPC.UTS.3070.00	pCurrentJob != NULL	g_PrinterContext.pCurrentJob = NULL g_PrinterContext.status = Wait
NPC.UTS.3070.01	pCurrentJob = NULL	g_PrinterContext.status = Wait
NPC.UTS.3080	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받아 충전이 요구되는 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Need Supplies로 변경한다.	
NPC.UTS.3080.00	Interfaces는 testing 제외	
NPC.UTS.3090	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받고, JOB WAIT LIST에서 최상위 작업을 불러와 출력을 시작한다. 출력 중인 상태를 LCD에 표시하고 PRINTER CONTEXT에 Status를 Printing으로 변경한다	

NPC.UTS.3090.00	전달 역할 등 단순한 프로세스 및 UI구성을 담당하는 프로세스는 테스트에서 제외	
NPC.UTS.3100	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받고, 출력중인 작업을 지속한다.	
NPC.UTS.3100.00	pCurrentJob.needPaper != pCurrentJob.printedCnt !feof(pCurrentJob.plnFile) == TRUE	g_PrinterContext.status = Printing Printing 출력
NPC.UTS.3100.01	pCurrentJob.needPaper != pCurrentJob.printedCnt !feof(pCurrentJob.plnFile) == FALSE	g_PrinterContext.status = Wait Wait 출력
NPC.UTS.3100.02	pCurrentJob.needPaper == pCurrentJob.printedCnt	fclose(pCurrentJob.plnFile) fclose(pCurrentJob.pOutFile) free(pCurrentJob) g_PrinterContext.pCurrentJob = NULL g_PrinterContext.status = Wait Wait 출력
NPC.UTS.3110	JOB WAIT LIST와 PRINTER CONTEXT를 Trigger로 전달받고, 잉크와 용지를 SUPPLIES DATA에서 Paper가 0이 아닐 때 10만큼, Ink가 0이 아닐 때 100만큼 감소 후 같은 량을PRINTER CONEXT의 Ink와 Paper에 증가시킨다.	
NPC.UTS.3110.00	g_PrinterContext.ink + refillInk > MAX_SUPPLIES_INK	refillInk = refillInk – (g_PrinterContext.ink + refillInk_MAX_SUPPLIES_INK)
NPC.UTS.3110.01	g_PinterContext.paper + refillPaper > MAX_SUPPLIES_PAPER	refillPaper = refillPaper – (g_PrinterContext.paper + refillPaper – MAX_SUPPLIES_PAPAER)
NPC.UTS.3110.02	(g_SuppliesData.Ink != 0 && g_PrinterContext.ink < MAX_SUPPLIES_INK) ‖ (g_SuppliesData.Paper != 0 &&	g_PrinterContext.status = Refilling

	g_PrinterContext.paper < MAX_SUPPLIES_PAPER)	
NPC.UTS.3110.03	(g_SuppliesData.Ink = 0 || g_PrinterContext.ink > MAX_SUPPLIES_INK) && (g_SuppliesData.Paper = 0 || g_PrinterContext.paper > MAX_SUPPLIES_PAPER)	g_PrinterContext.status = Wait

Table2. Test Case Identification

8.2 Test items

<Table 2. Test Case Identification 참고>

8.3 Input specifications

<Table 2. Test Case Identification 참고>

8.4 Output specifications

<Table 2. Test Case Identification 참고>

9 Testing tasks

Task	Predecessor tasks	Special Skills	Effort	Finish Date
(1) Unit Test Plan 작성	SRA 작성 SDS 작성 NPS 구현		3	2015-11-04
(2) Test Design Specification	Task 1	NPS에 대한 이해		
(3) Test Case Specification	Task 2	NPS에 대한 이해		
(4) Test Execution	Task 3	Test code 작성	2	
(5) Test Result Report	Task 4		1	2015-11-09
(6) 개발팀에 report 전달	Task 5		1	2015-11-10

Table 3. Testing tasks & Schedule

10 Environmental needs

C Language Compiler/Linker ex)Visual Studio 2010

11 Unit Test deliverables

12 Schedules

<Table 3. Testing tasks & Schedule 참고>